

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (Original) A method for accelerating storage access in a network, said method comprising:

- a) receiving a data record having a plurality of data segments;
- b) saving said data segments in a local memory of a network controller (NC);
- c) assigning a virtual write buffer (VWB) entry, in said NC local memory, for the incoming data record;
- d) reassembling said data segments of said data record using said VWB; and,
- e) sending said data record from the network controller directly to an I/O controller of a storage device.

2. (Original) The method of claim 1, further comprising:
allocating a private buffer in a host local memory.

3. (Original) The method of claim 1, wherein said NC is coupled to a storage target system and to a network.

4. (Original) The method of claim 2, wherein the data segments are virtually reassembled in said NC local memory to form a reassembled data record.

5. (Original) The method of claim 1, wherein said I/O controller is further coupled to a storage device.

6. (Currently Amended) The method of claim 1, wherein the data is received using a sub-process comprising:

- i) performing a transport layer processing on the data segments; and,

ii) assigning a memory object descriptor (MOD) ~~each~~ to each of the data segments.

7. (Original) The method of claim 6, wherein each said MOD points to a memory location where a corresponding data segment is stored in the NC local memory.

8. (Original) The method of claim 6, wherein said MODS are linked together to form a record structure.

9. (Original) The method of claim 2, wherein an available private buffer is used from a pool of pre-allocated private buffers.

10. (Currently Amended) The method of claim ~~4~~2, wherein said NC maintains a VWB table, wherein said VWB table includes at least a VWB entry.

11. (Original) The method of claim 10, wherein said VWB entry comprises at least two subentries, wherein a first sub-entry is an offset field and a second sub-entry is a pointer field.

12. (Currently Amended) The method of claim 11, wherein the memory address space of said VWB entry is mapped to the memory address space of the allocated private buffer when the VWB entry is assigned.

13. (Original) The method of claim 11, wherein reassembling said data segments comprises setting said offset field and said pointer field.

14. (Original) The method of claim 13, wherein setting said offset field and said pointer field further comprises:

i) iteratively, for each MOD, determining a size of a corresponding data segment pointed by said each MOD;

ii) setting said offset field to a size of said corresponding data segment pointed by said MOD; and,

iii) setting said pointer field to point to said each MOD.

15. (Original) The method of claim 14, wherein a VWB entry is associated with each said allocated private buffer.

16. (Original) The method of claim 15, wherein the reassembled data record is sent to the I/O controller using a sub-process comprising:

- a) providing said I/O controller with an address space of said private buffer associated with said VWB entry;
- b) translating the address space of said VWB entry to a physical address location of said reassembled data record;
- c) obtaining said reassembled data record from said physical address location; and,
- d) sending said reassembled data record directly to said I/O controller over an I/O bus.

17. (Original) The method of claim 16, wherein said physical address location designates a location of said reassembled data record in the NC local memory.

18. (Original) The method of claim 16, wherein said I/O controller is provided with the address of said private buffer, upon initiating a direct memory access (DMA) request by said I/O controller.

19. (Original) The method of claim 16, wherein the NC sends said reassembled data record, upon a reception of a DMA read request initiated by said I/O controller.

20. – 38. (Cancelled)

39. (Original) A network controller (NC) capable of accelerating storage access, comprising:

- a core processor adapted to execute an accelerated storage access process ;
- a receive handler adapted to receive data record from a network;

a direct memory access controller (DMAC) adapted to transfer said data record directly to an I/O controller using a virtual write buffer (VWB); and,
a local memory.

40. (Original) The network controller of claim 39, wherein said NC is coupled to a storage target system and to said network.

41. (Original) The network controller of claim 40, wherein said network is at least one of: network attached storage (NAS), storage area network (SAN), system area network (SAN).

42. (Original) The network controller of claim 39, wherein said I/O controller is coupled to a storage device.

43. (Original) The network controller of claim 39, wherein data segments constituting said data record are virtually reassembled in said NC local memory.

44. (Original) The network controller of claim 43, wherein the receiver handler is adapted to receive the data record comprising of a plurality of data segments and being assigned a virtual write buffer (VWB) in a VWB table, said VWB being adapted to enable reassembling the data segments to form an reassembled data record.

45. (Original) The network controller of claim 44, further adapted to allocate a private buffer in a local host memory.

46. (Original) The network controller of claim 45, wherein the NC is adapted to perform a transport layer processing on said data segments and assign a memory object descriptor (MOD) each to each of said data segments.

47. (Original) The network controller of claim 46, wherein said MOD points to a memory location of said segment in the NC local memory.

48. (Original) The network controller of claim 46, wherein said MODS are linked together to form a record structure.

49. (Original) The network controller of claim 45, wherein an available private buffer is used from a pool of pre-allocated private buffers.

50. (Original) The network controller of claim 45, wherein said NC maintains a VWB table, wherein said VWB table includes at least a VWB entry.

51. (Original) The network controller of claim 50, wherein said VWB entry comprises at least two sub-entries, wherein a first sub-entry is an offset field and a second sub-entry is a pointer field.

52. (Original) The network controller of claim 51, wherein memory address space of said VWB entry is mapped to memory address space of the allocated private buffer when the VWB entry is assigned.

53. (Original) The network controller of claim 52, wherein data segments are reassembled by setting the offset field and the pointer field.

54. (Original) The network controller of claim 53, wherein the offset field is set to a size of a data segment pointed to by each MOD and the pointer field is set to point to each MOD.

55. (Original) The network controller of claim 45, wherein a VWB entry is associated with each said allocated private buffer.

56. (Original) The network controller of claim 55, wherein the NC is adapted to provide the I/O controller with an address space of the private buffer associated with the VWB entry, said address space corresponding to a physical address location of the designated data record, the

NC being further adapted to obtain the reassembled data record from the physical address location and send the reassembled data record to the I/O controller.

57. (Original) The network controller of claim 56, wherein said physical address location designates a location of said reassembled data record in the NC local memory.

58. (Original) The network controller of claim 57, wherein said I/O controller is provided with the address of said private buffer, upon initiating a direct memory access (DMA) request by said I/O controller.

59. (Original) The network controller of claim 58, wherein the NC sends said reassembled data record, upon a reception of a DMA read request initiated by said I/O controller.